**TRABAJO PRACTICO**



NOMBRE: SIRIA BARRIGA VILLARROEL

Nº REGISTRO:219215944

CARRERA: ING. INFORMATICA

MATERIA: PROG. DE APLICACIÓN DE TIEMPO REAL

AÑO: 2023

**Lista de 5 Sistemas de Control de Versiones (VCS), con sus ventajas y desventajas**

**SVN**



SVN, a diferencia de Git, se basa en una arquitectura cliente-servidor. Presenta algunas funcionalidades interesantes, como versiones de enlaces simbólicos, soporte para directorios vacíos y registros de operaciones de copiar, mover, borrar y renombrar. **Un gran sistema para Windows y proyectos que no requieran acceso distribuido.**

* **Ventajas:**
  + Fácil de configurar y administrar.
  + Muy buen soporte para Windows.
  + Integración excelente con los principales IDE y las herramientas ágiles.
* **Desventajas:**
  + Al basarse en una arquitectura cliente-servidor, todas las versiones deben almacenarse en el mismo lugar.
  + Generación de ramas más compleja.
  + Repositorios más pesados.
  + Sin información sobre la hora de modificación de los archivos.
  + Sin soporte para versiones firmadas.

**Mercurial**



Mercurial, igual que Git, es un sistema de control de versiones d**istribuido ideal para usuarios poco experimentados o perfiles poco técnicos**.

* **Ventajas:**
  + Intuitivo y sencillo de manejar, con una curva de aprendizaje mucho más corta.
  + Veloz, potente y ligero.
  + Manejo sólido de archivos de texto plano y, también, binarios.
* **Desventajas:**
  + No funciona bien con extensiones.
  + Presenta menos funcionalidades que Git.
  + Solo soporta complementos programados en Python.

**CVS**



CVS se basa en una arquitectura cliente-servidor como la de SVN, lleva presente mucho tiempo en el mundo del desarrollo y **es una opción excelente para los proyectos de código abierto.**

* **Ventajas:**
  + Muy buena comunidad para la resolución de dudas y búsqueda de soluciones.
  + Gran soporte multiplataforma.
  + Línea de comandos robusta y útil para el uso de scripts.
* **Desventajas:**
  + No soporta revisiones firmadas, como SVN.
  + Poco soporte para el control de código fuente distribuido.
  + No contempla *checkouts* ni *commits* atómicos.

**Monotone**



**Monotone es una herramienta de control de versiones distribuida de código abierto que puede integrarse en multitud de sistemas operativos**. Prima la integridad ante el rendimiento, emplea técnicas criptográficas para rastrear las versiones y autenticaciones de archivos, puede importar proyectos CVS y utiliza el protocolo [netsync](https://www.netsync.com/).

* **Ventajas:**
  + Excelente funcionamiento de la ramificación.
  + Buena documentación oficial.
  + Requiere poco mantenimiento.
  + Sencillo, con una curva de aprendizaje corta.
  + Interfaz gráfica intuitiva.
* **Desventajas:**
  + Problemas de rendimiento en algunas operaciones.
  + No permite hacer *checkout* ni *commit* detrás del proxy.

### TFS



TFS, un acrónimo de team foundation server es un producto de control de versiones de Microsoft. Se basa en un modelo de repositorio distribuido cliente-servidor y tiene una licencia propietaria. Proporciona compatibilidad con sistemas operativos multiplataforma de Windows a través de Visual Studio Team Services (VSTS).

* **Ventajas**
* Fácil administración. Interfaces familiares y estrecha integración con otros productos de Microsoft.
* Permite la integración continua, el equipo construye y la integración de pruebas unitarias.
* Gran soporte para operaciones de ramificación y fusión.
* Políticas de registro personalizadas para ayudar a implementar una base de código estable y estable en su control de código fuente.
* **Desventajas**
* Conflictos frecuentes de fusión.
* Siempre se requiere conexión al repositorio central.
* Bastante lento para realizar operaciones de extracción, registro y bifurcación.